

A system and method to eliminate the critical path in a processor-based system by sending a signal to transition the memory and I/O control unit to the read/write state before instruction decode completes up to the end. The decode stage of the operation code of the instructions, if it is made clear that the step of reading and writing in which the memory or I/O device must be accessed should be executed, the processor immediately sends a read/write request to the memory control unit and the I/O control unit before decoding the remaining part of the instruction. Once the remaining part of the instruction has been decoded, if it has been decided that the storage place that is accessed is in either the memory or the I/O device, cancellation processing occurs. In this cancellation processing, if the storage place that is accessed is within the memory, the I/O unit transitions from the read/write state to the idle state. However, if the access destination is the I/O device, the memory control unit transitions from the read/write state to the idle state.

特表平7-505242

第6部門第3区分

(43) 公表日 平成7年(1995)6月8日

(51) Int. Cl. *	識別記号	庁内登録番号	F I
G 0 6 F 9/38	3 5 0 2	9193-5B	
9/34	3 5 0 A	9189-5B	
12/00	5 6 0 C	9366-5B	

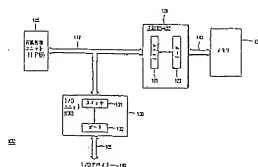
帯込請求 請求項 予備審査請求 有 (全 10 頁)

(21) 出願番号	特願平5-515523	(71) 出願人	セイコーエプソン株式会社
(26) (22) 出願日	平成5年(1993)3月2日		東京都新宿区西新宿2丁目4番1号
(85) 翻訳文提出日	平成6年(1994)9月2日	(72) 発明者	ミヤヤマ ヨシユキ
(86) 国際出版番号	P C T / J P 9 3 / 0 0 2 5 9		アメリカ合衆国 95050 カリフォルニア
(87) 国際公開番号	W O 9 3 / 1 8 4 5 1		州 サンタ クララ、ランチョ マグコー
(87) 国際公開日	平成5年(1993)9月16日		ミック ブルバード 2171
(31) 優先権主張番号	8 4 6 . 2 3 1	(72) 発明者	ダン テンネーロン
(32) 優先日	1992年3月6日		アメリカ合衆国 95131 カリフォルニア
(33) 優先権主張国	米国 (US)		州 サン ホセ、リビシ ウェイ 1915
(81) 指定国	J P	(74) 代理人	弁理士 鈴木 喜三郎 (外1名)

(54) 【発明の名称】 メモリ制御ユニット並びに入出力制御ユニットの動作におけるクリティカル・バスの制御

## (57) 【要約】

命令のデコードが最後まで完了してしまう前に、メモリやI/O制御ユニットを読み出し/書き込み状態に移すために信号を送ることによって、プロセッサ・バスのシステムにおいてクリティカル・バスを制御するためのシステム並びに方法。命令の演算コードのデコード段階が、メモリまたはI/Oデバイスがアクセスされなければならない読み出し書き込みのステップが実行されるべきであることを明らかにすれば、プロセッサは、命令の残りの部分をデコードする前に、直ちに読み出し書き込み要求をメモリ制御ユニット及びI/O制御ユニットに送る。一旦命令の残りの部分がデコードされ、アクセスする記憶場所がメモリまたはI/Oデバイスのどちらにあることが決定されれば、取消し処理が起きる。この取消し処理において、アクセスする記憶場所がメモリ内ならば、I/Oユニットは読み出し書き込み状態からアイドル状態に移す。しかしながら、アクセスの宛先がI/Oデバイスならば、メモリ制御ユニットは読み出し書き込み状態からアイドル状態に移す。



## 指示の形式

1. 読み出し/書き込み動作中にプロセッサ・ベースのシステムにおいてタリチアル・バスを断続する方法で、前記システムがメモリ解放ユニット(MCU)、入出力制御ユニット(I/OU)、並びに命令集合をそれぞれ1個ずつ含む。解放方法が、

(1) 命令集合から命令の第1の部分をデコードするステップと、

(2) 前記の命令が読み出しと書き込みのうちの少なくとも1つを命じているか否かを決定するステップと、

(3) ステップ(b)で読み出しと書き込みを命じていることが確定されれば、読み出し/書き込み状態に入るために要求されるMCUとI/OUの両方に送るステップと、

(4) 読み出し/書き込み状態によってアクセスされるべきアドレスを決定するために命令の残りの部分をデコードするステップと、

(5) アクセス対象のアドレスをデコードするステップで、それは、そのアクセスの対象がメモリ(I/Oデバイス)かを決定するためにMCUとI/OUの少なくともどちらか一方によって実行されるデコーディングであること、さらに、

(1) アクセス対象のアドレスをデコードするステップで、それが前記の少なくとも1個以上の装置のうちのどれがアクセスされるべきかを決定するために少なくとも1個以上の装置によって実行されるデコーディングであること、さらに、

(2) 前記のデコードするステップ(1)による結果に基づいて要求の残りの部分を取り扱うステップと、

から成り立つことを特徴とする方法。

5. 前記ステップ(1)及び(4)が並列で実行されることを特徴とする請求項4記載の方法。

6. 前記ステップ(c)における要求が同時に送られることを特徴とする請求項4記載の方法。

7. 装置へのアクセス動作中にプロセッサ・ベースのシステムにおいてタリチアル・バスを断続するたりのシステムで、前記システムが多量の装置並びに1個の命令集合を含み、さらに

(1) 命令集合から命令の第1の部分デコードするための手段と、

(2) 前記の命令がアクセス動作を命じているか否かを決定するための手段と、

(3) ステップ(b)で装置へのアクセスが求められていることが確定すれば、アクセス状態に入るために少な

(1) 前記のデコードするステップ(1)による結果に基づいてMCUとI/OUに対する要求の装置の一つを取り扱うステップと、

から成り立つことを特徴とする方法。

2. 前記ステップ(4)及び(4)が並列で実行されることを特徴とする請求項1記載の方法。

3. 前記ステップ(c)における要求が同時に送られることを特徴とする請求項1記載の方法。

4. 装置へのアクセス動作中にプロセッサ・ベースのシステムにおいてタリチアル・バスを断続する方法で、前記システムが多量の装置並びに1個の命令集合を含み、解放方法が、

(1) 命令集合から命令の第1の部分デコードするステップと、

(2) 前記命令がアクセス動作を命じているか否かを決定するステップと、

(3) ステップ(b)で装置へのアクセスが求められていることが確定すれば、アクセス状態に入るために少なくとも2個以上の装置に要求を送るステップと、

(4) アクセスされるべきアドレスを決定するために命令の残りの部分をデコードするステップと、

(1) 2個以上の装置に要求を送る手段と、

(2) アクセス対象のアドレスを決定するために命令の残りの部分をデコードする手段と、

(3) アクセス対象のアドレスをデコードする手段で、それはデコーディングが前記の少なくとも2個の装置のうちのどれがアクセスされるべきかを決定するために少なくとも1個以上の装置によって実行されるデコーディングであること、さらに、

(4) 前記のデコードするステップ(1)による結果に基づいて要求の残りの部分を取り扱うための手段と、

から成り立つことを特徴とするシステム。

8. 決定の前記手段が、前記命令が読み出し及び書き込み動作の少なくともどちらか一方を命じているかどうかを決定する手段であり、

送らたりの前記手段が、ステップ(b)において読み出しと書き込みが求められていることが確定すれば、読み出し/書き込み状態に入るためにMCUとI/OUの両方に要求を送る手段であり、

命令の残りの部分をデコードするための前記手段が、

読み出しと書き込み動作によってアクセスされるべきアドレスを決定するために、命令の残りの部分をデコードするための手段であり、

アドレスをデコードするための脱記手段がアクセス対象のアドレスをデコードするための手段であり、それはアクセスの対象がメモリかI/Oデバイスかを決定するためにMCUまたはIOUが少なくともどちらか一方によって実行されるデコーディングであること、さらに、

脱記手段がアドレスをデコードするための脱記手段による結果に基づいてMCUとIOUへの脱記要求の1つを取り除くための手段であることを特徴とする請求項3記載のシステム。

## 明 細 書

### 発明の名称

メモリ制御ユニット並びに入出力制御ユニットの

動作におけるクリティカルパスの排除

(Elimination of the Critical Path in Memory Control Unit and Input/Output Control Unit Operations)

### 発明の要旨

#### 1. 要旨上の利用分野

本発明は一般的にはマイクロプロセッサ・ベースのシステムの分野に關し、より具体的には、少なくとも2つ以上のサブシステムに組み出し/書き込み(READ/WRITE)要求を同時に送り、引き続いて不要な要求を取り除くためのシステム並びに方法に關する。

#### 2. 関連技術

典型的なコンピュータ・ベースのマイクロプロセッサ・システムは、3次サブシステムから成り立っている。それらは、即ち、1番の基盤は、少なくとも1個以上の内部処理ユニット(IPU)、そして1つの入出力(I/O)サブシ

ステムである。従来のマイクロプロセッサ・ベースのシステムの中には、性能を上げるためにパイプライン方式として知られる方法を用いるものがある。パイプライン処理では、命令実行の順みをステップ(例えば、フェッチ、デコード、及び実行)はパイプライン・ステージと呼ばれる独立した単位によって実行される。そのステップは、アクセスが一度に少なくとも1つ以上の命令を処理できるように、ほぼ等間隔で実行される。

パイプラインのスケジューリング、どの位の程度で1つの命令がパイプラインから出る(exit)かによって決定される。パイプライン・ステージは相互に繋がっているから、全てのステージが同時に実行するように制御されなければならない。しかしながら、実際には、フェッチ、デコード、実行のステージの中には他に比べて時間が掛かるものもある。そのために、其の遅延性が実行されているとは言えない。1サイクル時間に割り当てられる処理時間は最低ステップを実行するのに掛かる時間である。言い換へ、そのステップを複数のサイクル分間で実行することである。いずれの場合も、システムの効率性は最低ステップによって上げられてしまう。

パイプライン方式の非効率については、Hennessy他著の「Computer Architecture a Quantitative Approach」(Morgan Kaufmann 出版社、1980)を参照のこと。

ソース・プログラムの動作中、IPUは立派な遅延をもく、I/O処理のデータにアクセスすることが必要になる。具

して、IPUは、プロセッサのサイクル時間が数ナノ秒の、マイクロプロセッサ・ベースのシステムで最も遅い要素である。その一方、記憶体でI/Oデバイスのサイクル時間は数ナノ秒である。従って、マイクロプロセッサの性能を制約する要因の一つはメモリ・アクセスの遅延ということになる。

従来のシステムにおいては、命令デコードの初期の段階から、プロセッサI/Oまたはメモリへのアクセスが起きるかどうかがわかっている。しかしながら、命令が読めず後述まで完全にデコードされるまで、アクセス対象が記憶体なのかそれともI/Oなのかのシステムは決定されない。従来のシステムでは、読み出し/書き込み状態に遷移するためのコマンドとメモリ制御ユニット(MCU)またはI/O制御ユニット(IOU)に送る前に、OPCODE(演算コード)及びレジスタ・アドレス解析を各自の命令単位でデコードする。言い換えれば、従来のシステムはこうした処理を逐次に実行し、命令の完全なデコードのステップ、MCUまたはIOUを読み出し/書き込み状態に遷移するステップは「クリティカル・パス」上にある。その結果、従来のシステムは、メモリやI/Oデバイスに実際にアクセスするという時間のかかるステップに入る前に、命令を逐段までデコードし終るのを待つ必要な時間を浪費しているのである。

典型的なクロックタイミングにおいて、システム・クロックはスLEEP状態として知られる状態がある。

MCUやIOUなどのスレーブ・デバイスが、読み出し／書き込みなどの動作を実行するためにバスへのアクセスが可能とされるのは、このスレーブ状態の間だけである。従って、クロックのサイクル毎に、アクセス対象のデバイスが現在のスレーブ・サイクルに間に合うように読み出し／書き込み状態に入れるように、クリティカル・バス上の全ての処理が実行されるのに十分な長さでなければならぬ。クリティカル・バス上の処理が全てその時間内では実行しきれないようにクロックのサイクル毎に処理が遮断されていれば、読み出し／書き込み動作は今後のスレーブ・サイクルに間に合わず、その次のスレーブ・サイクルまでさらにクロック同期待たなければならぬ。従って、必要なのは、アクセス対象の実際のアドレスがデコードされるのを待たずに、OPCODEがデコードされ実行プロセッサが直ちにメモリに対して読み出しまたは書き込み要求を出せるようにするシステム並びに方法である。こうすることによって、読み出し／書き込み要求をクリティカル・バスからパシベル・バスに移すことによってクリティカル・バスの長さが短縮される。この方法で、システムのクロック・レートが速まると、メモリ及びI/Oサブシステムの実際のレイテンシが短縮される。

図1のフローでメモリがアクセス対象であったことが明らかになると、IOUはアイドル状態に戻る。一方、図1のフローでI/Oデバイスがアクセス対象であったことが明らかになれば、MCUはアイドル状態に戻る。

## 図1の図説

本発明はプロセッサ・ベースのシステムに利用可能なリソースと、バスを確保するためのシステム並びに方法を提供するものである。本発明に基づく方法においては、命令の選択コードがデコードされ、その選択コードがシステム・メモリまたはI/Oデバイスにアクセスを命じる動作を呼び出すと、読み出し／書き込み要求が直ちにプロセッサ・ベースのシステムのメモリ制御ユニットにCPUと入出力制御ユニットIOUの両方に送られる。プロセッサが命令の残りの部分を実行すると同時にこの読み出し／書き込み要求は送られるため、現在のクロックが終わる前にMCUとIOUがアイドル状態から読み出し／書き込み状態に移行することが可能になる。

この方法によって、サイクル毎に、プロセッサが現在の命令をデコードするのにかかる合計時間のみに抑えられるようになる。さらに時間を追加することなく現在のクロック・サイクル毎に、MCU並びにIOUは読み出し／書き込み状態に移行できる。それは勿論ならば、MCUとIOUは既にその状態に入っているからである。

命令が最後まで残らずデコードされ、メモリ・アクセスまたはI/Oアクセス用のアドレスが計算され、さらにMCUまたはIOUの宛先がデコードされたら、取崩し処理が行われる。取崩し処理では、MCUまたはIOUのいずれかが読み出し／書き込み状態からアイドル状態に戻る。

## 図1の図説を説明

本発明については他の特許請求の範囲で具体的に説明されている。今までに述べたさらにそれ以外の本発明の様々な点については、他の図説と共に以下に要述する説明を参照することによってより理解が深まるであろう。

第1図は本発明に基づくシステム・アーキテクチャ100の概略ブロック図である。

第2図はシステム内メモリ・マップである。

第3図は典型的な命令フォーマットを説明したものである。

第4図は、2個のレジスタに対する1つの演算の処理を通して1つのアドレスが計算される典型的な命令フォーマットを説明したものである。

第5図はクリティカル・バス上の全ての演算を含む従来のシステムの状態遷移の概略を示す図である。

第6図は本発明の方法に基づく例を示す状態遷移図である。第6図は本発明によるIOUの取崩し手続のためのタイミング図である。

第7図は本発明の方法を示す状態遷移図である。

## 図1

第1図は本発明が適用できるアーキテクチャを代表する。マイクロプロセッサのアーキテクチャ100の最低レベルを示したものである。第1図において、本発明の好適な実施例に基づいて、100と指定されたマイクロプロセッサ・システムのアークテクチャが提示されている。システム・アーキテクチャ100は、内部プロセッサ・ユニット110、メモリ制御ユニット120(MCU)130、I/O制御ユニット140(OU)150、主記憶160、及び外部I/Oデバイス180から成る。

主記憶160は、好適な実施例においては、32ビット幅のメモリ・バス140によってMCU120に接続されている。同時に、I/Oデバイス180は、好適な実施例においては、32ビット幅の外部I/Oバス135によってOU130に接続されている。

MCU120はIPU102と主記憶130間の読み出しきき込み動作を制御する回路である。本発明に基づく好適な実施例におけるMCU120は、スイッチ・ネットワーク121及びポート122から成っている。本明細書に記載の例においてはポート122は16ビットしか示されていないが、本発明に基づく好適な実施例は複数のポート122を備えている。

また、IPU102とポート132の要求を調整する手段としても扱われる。ポート132(単独または複数)は、I/Oデバイス180にデータを送り、且つそこからデータを受け取る役目を果たす。

第2図はシステム・メモリの構成を示したものである。好適な実施例において、メモリ構成は2階層に分類される。I/Oバス領域210とメモリ・バス領域220がある。この構成の結果、システムは主記憶160とI/Oデバイス180の両方に対する読み出しきき込み動作を要求することができるようになる。読み出しきき込み動作の動作を含む命令は、システムのメモリ・マップ200上に容易に可能なアドレスを含むことになる。

本発明に基づく好適な実施例において、命令加重は3ステージのバイザインに分類されている。その3ステージとは、フェッチ、デコード、及び実行である。最初、命令はIPU105によってメモリからフェッチされる。その次に、命令は指示された階層(例えば、論理、算数、格納)を確定するためにデコードされる。3番目且つ最後は、そのデコードされた命令はその機能の通りに実行される。

命令をデコードしている時、レジスタ値及びアドレス指定モード値は命令サイズに大きな影響を及ぼす。これはアドレス指定モードのフィールドとレジスタ・フィールドが1つの命令の中で同時に扱われることが知られており、事実上ほとんどの命令が、演算コードの指定よ

うにスイッチ・ネットワーク121はMCU120とIPU103間の通信インフラを可能にする。スイッチ・ネットワーク121はまた、ポート122の要求を調整する手段としても扱われる。ポート122(単独または複数)は、主記憶130にデータを送り、且つそこからデータを受け取る役目を果たす。

スイッチ・ネットワーク121の役目は、IPU105及びOU130からの様々な命令並びにデータ要求を受け取ることである。IPU105及びOU130はメモリ要求と称げられることがある。要求を受け取り調整した後に、要求は対応するメモリ・ポート132に渡される。対応するポート132の状況は命令アドレスの階層によって変化する。ポート132(単独または複数)は次に必要なタイミング信号を生成し、外部メモリ150のアドレス指定された部分にデータを送るか、またはそこからデータを受け取るかとする。

OU130はIPU105とI/Oデバイス180間の読み出しきき込み動作を制御する回路である。本発明に基づく好適な実施例におけるOU130は、スイッチ・ネットワーク131及びポート132から成っている。本明細書に記載の例においてはポート132は16ビットしか示されていないが、本発明に基づく好適な実施例は複数のポート132を備えている。MPUの命令と同時に、OU130のスイッチ・ネットワーク131はOU130とIPU105間の通信インフラを可能にする。スイッチ・ネットワーク131

はアドレス指定モードやレジスタ・フィールドのコード化に必要にビットをとっている。このように、アドレス指定モードとレジスタ・フィールドが各ビットの大きな割合を占めるため、それらのコード化は即座に命令のデコードを遅延させるかにかた(影響)を与える。

命令は互つかのセクションに分類されることが多い。第3A図は命令の一部とその構成要素を示すものである。第3A図において、命令300はOPCODE310を含んでいる。OPCODE310は、一旦デコードされれば、どのオペレーションを実行すべきかプロセッサに伝える。オペレーションにはADD(加算)、COMPARE(比較)、CLEAR(クリア)などの算術演算、BRANCH(分岐)やJUMP(飛び越し)などの制御動作、RETURN FROM INTERRUPT(割り込みからの戻り)などのシステム動作、並びにMOVE(移動)やPUSH(プッシュ)などのデータ転送動作などがあろう。

レジスタやメモリに100デバイスのそれだけのアドレスは、必要に応じて、プロセッサにどの記憶場所に於いてオペレーションを行なうべきか伝えるために、命令の中に包含されている。これらは記憶場所320、330で命令にコード化される。例えば、OPCODE310でADDの演算を行なうようにプロセッサに命令すれば、プロセッサは、加算対象の読み出しアドレスを決定するために、記憶場所320と330を足し合わせることになっている間接を

デコードする（直書きアドレス指定の場合）。

命令300の実際のフォーマットはシステム・アーキテクチャによって異なる。どうすれば本説明に基づく仮定が他の命令フォーマットに適用できるかは当業者には明かになるであろう。

第38図はもう少し少し複雑なアーキテクチャ用の命令フォーマットを示したものである。同38番におけるOPCODE310は、第3A図に示したフォーマットにおいて説明したのと同じオペレーションを行なうように、プロセッサに命令することである。しかしながら、ここでは、アドレスの計算が異なる。このフォーマットでは、最初のアドレスは2つのアドレスをデコードし計算することによって計算される。最初のアドレスを計算するために、プロセッサはRindex340とRbase350をデコードする。実際のアドレスは指定するなかにこれらの記憶場所における値に所定のオフセットを加算される（この場合、合計される）。このタイプのアーキテクチャは多数のプログラマに有益である。2番目のアドレスを計算する必要がある場合、記憶場所Raddress360における値がデコードされるだけである。

バイタインの命令アコード部分は幾つかの段階に分けられる。命令のそれぞれ異なる部分、即ち、OPCODE310、Rindex340、Rbase350、Raddress360はそれぞれ段階でデコードすることができる。従って、プロセッサは、命令の残りの部分に先行してOPCODE310を

先にデコードすることができる。一旦命令のOPCODE310部分がデコードされれば、プロセッサは命令がどんな動作を命じようとするかわかる。

従って、例えば、命令は計算または比較であるとデコードされたOPCODE310がIFU102に伝えられ、プロセッサは演算対象値に他の記憶場所からアクセスしなければならぬことを知る。第38図と第39図において、従来のシステム用の非従属的なシリアルについて説明している。命令はフェッチされ、OPCODE310がデコードされる。OPCODE310がプロセッサにロードし格納動作を演算するように命令すれば、IFU105は格納対象の値のある記憶場所を指定するためにRindex340とRbase350をデコードする。IFU103は次にRindex340とRbase350をデコードする。一旦デコードされれば、Rindex340とRbase350はデータの最終アドレスを計算するのに関与して送れる。その次に、この最終アドレスはデコードされるためMCU130またはIOU130に送れる。デコードされたら、MCU120またはIOU130のいずれか読み出し/書き込みを実行すべき方が、読み出し/書き込み状態に遷移される。しかしながら、こうした状態を全て逐次に実行し、しかも今度のスレーブ・クロック・フェーズに間に合うようにそれに対応するユニットを読み出し/書き込み状態に遷移させるには、現在のクロック・フェーズ内で時間的に十分とはいかない場合がある。スレーブ・クロック・フェーズはその期間読み出し/書き込み

動作が発生しうる唯一のフェーズであるから、現在のスレーブ・フェーズに間に合うを付けば、今のシステムは次のスレーブ・フェーズまでクロック期間をまるまる待たなければならぬ。従って、これらの状態は全て逐次に実行され、それら毎度クロック・レートでは行列の読み出し時間よりずっと長くなる。スレーブ・フェーズに間に合わない状態となるから、クリティカル・パスが存在しているということになる。或いは、パス上を流れてフェーズが終了するように、システム・クロックのサイクル時間を短くしなければならない。しかしながら、この代替案は、遅延を遅くすることになるから、望ましい問題解決法とは言えない。

第40図は従来のシステムに関する上記のロード/格納のシナリオに基づいたタイミング図並びに状態図を示したものである。予期の動作時間は例として記述するものである。第40図、第41図のシナリオは命令格納と他の関連で演算時間は異なる。第40図において、第1クロック期間T0T1で、ロード/格納命令がフェッチされる。第2クロック期間T2T3で、命令がデコードされる。第3OPCODE310がデコードされ、その後Rindex340、Rbase350、Raddress360がデコードされる。Rindex340及びRbase350はT3のクロック・フェーズでデコードされる。この処理はおよそ5ナノ秒ばかり、アドレス410で示されている。この処理が完了すると、アドレス計算が行われる。この計算に要する時間はプロ

セッサに示すように7ナノ秒である。アドレス計算実行後には、アドレスはMCU120もしくはIOU130によってデコードされなければならない。ブロック430で示した、このデコーディングにかかる時間は各例で20ナノ秒である。

アドレスがブロック430でデコードされた後、MCU130またはIOU130のいずれか対応する状態ユニットは、次のスレーブ状態に読み出し/書き込み状態に入っていないなければならない。第40図において、T4、T6、T8などはスレーブ状態。即ち、メモリ110の読み出し/書き込みが起きる唯一の状態である。命令とアドレスのデコーディング動作がその後のT3でクロック周期の20ナノ秒を過ぎ、且つ既に今度のスレーブ・フェーズT4で読み出し/書き込みが起きるようにしたいため、クロック・フェーズの間隔は30ナノ秒より長くなければならない。20ナノ秒を超えてさらに要する時間はMCU120またはIOU130の対応する状態ユニットを読み出し/書き込み状態に遷移するのにかかる時間である。そのフェーズの時間が過ぎれば、MCU120またはIOU130はT4のクロック・フェーズに間に合うように読み出し/書き込み状態に入る必要がある。クリティカル・パスが存在し、さらにシステムは読み出し/書き込み動作を実行するためにT6のクロック・フェーズに入るまで待たなければならない。しかるに、従来のシステムの欠陥は、命令アコード動作及びメモリリダイレクトIOサブイ

スの読み出し／書き込み状態への遷移が逐次起こり、且つ両端クロック・レートではクリティカル・パスを形成することである。その結果、クロック・サイクル毎回の増分(クロック・レートの倍々)を繰り返したり、その増分のみ出し／書き込み動作を遅延しおこなはねならない(増分の無効化)のクロック・増分に際する必要がある。この様な結果は、命令はさらにクロック・サイクル増えないと、即ち永続的ブレイクまで、動作完了しないということである。

## 図例

本説明では、アドレス・デコード動作に先行してMCU110及びIOU130を読み出し／書き込み状態に遷移することによって、このクリティカル・パスの突進を回避する。言い換えれば、アドレス増分から読み出し／書き込み状態への遷移は、アドレスがクリティカル・パスから取り除かれるということである。第3図は本説明に基づく動作を例示したタイミング及び状態の図である。第3図および3図において、デコードされたOPCODE130がメモリまたはIOデバイスに対する読み出し／書き込み動作が起る限りは示さないことを示してあれば、IPU103の読み出し／書き込み要求320はOPCODE130のデコード直後からMCU120及びIOU130に送達して送られる。IPU103は他のレジスタがデコードされるまで待つこと

はしない。この方法では、命令の残りの残高デコードされている間に、MCU120及びIOU130はT3のクロック増分中にアイドル状態340から脱し、読み出し／書き込み状態に入る。従って、読み出し／書き込み状態に遷移する動作に遅延を招かぬから、システムT6のフェーズ増分まで待たされることにならない。即ち、クリティカル・パスがなくなる。やがて、MCU120及びIOU130は、スレーブのクロック・フェーズT4発生より遅に読み出し／書き込み状態330に入っている。

この概念に、従来のシステムについて元に変更したのと同じタイミングの概念を使っておくことができる。例えば、レジスタのデコード・タイミング、アドレス増分ならはアドレスのデコード・タイミングに異なる増分がT0より少し下ならば、遅延T0より少し下のクロックを他のシステムでクリティカル・パスが存在しない。それは可能である。IOU120及びMCU120はT4の発生直前に既に読み出し／書き込み状態に入っているからである。

図3がデコードされ、MCU120とIOU130のどちらが必要なのかが分かれば、IOUの増分は信号360が生成され、従って不要の増分ユニットをアイドル状態に戻す。

増分380及び3図において、OPCODE130のデコードは、クロック310で示すように、T2、T3のクロック増分の増分に発生する。このデコードが完了した直後に、IPUの読み出し／書き込み要求320がMCU120とIOU130に送られる。要求320はMCU120を読み出し／

書き込み状態330に、IOU130を読み出し／書き込み状態330にする。読み出し／書き込み状態330に共にT4のスレーブ・クロック・フェーズ間の増分に似合うように既に発生している。

読み出し／書き込み要求と同期的に、命令の残りの増分が、対応するアドレスを指定するためにデコードされる。Rindex340及びRindex330のデコード・タイミングはプロトタイプ12に依存する。アドレス対象のアドレスの増分。即ちRindex340とRindex330への加算はプロトタイプ14で示す通りである。

増分ユニットMCU120またはIOU130の一方が読み出し／書き込み動作を実行し、他方はしない。プロトタイプ330に示すように、アドレス・デコードが終了するまでどちらがどちらのかわからない。アドレス・デコードが完了した後は、IOUの増分は信号360が生成され、従って読み出し／書き込み動作を行ってはいないユニットを取り出す。このユニットはアイドル状態340に戻る。

第3図に示した図において、アドレス対象のアドレスは、たとえばメモリ内に存在している。従って、そのアドレスがデコードされると、IOデバイスにはアクセスされるべきでないことが分かる。次に増分は信号360が生成され、IOU130はアイドル状態340に戻る。MCU120は読み出し／書き込み状態330のままで、メモリ・アクセスが完了する。

増分を両端に於いて、MCU120はIPU103に与

て増分されたアドレスをデコードする。アドレスがデコードされ、アドレスがメモリ330とIOU130のどちらに属しているかが判明すれば、MCU130は自分を読み出し／書き込み要求を取り出しアイドル状態340に戻る。もしくは増分は信号360がIOU130送るかのいずれかを行なう。このアドレス・デコードはMCU120またはIOU130のいずれか、或いはその両方に与えられなければならないことは明かである。

本説明に基づく従来のタイミング図は第4図に示す通りである。第4図において、IPU、増分信号320はMCU120及びIOU130の両方に送られる。MCU120は増分信号410Aに反応し、IOU130は増分信号410Bに反応する。MCU120とIOU130は共に、MCU\_STATE120A及びIOU\_STATE620Bが示すように、読み出し／書き込み状態にある。一旦アドレスがデコードされれば、MCU120またはIOU130のいずれか不正のユニットを取り出すために、MCU/IOU\_CAREL増分が生成される。第4図はIOU130が示す。従ってIOU\_STATE120Bがアイドル状態に増分を指示したものである。

第7図はMCU120とIOU130の状態を示すものである。増分及び3図において、両端は増分デコードを生成するために使われる状態マシンが示されている。増分は増分はMCU120とIOU130にもある。T4でIPU103からの読み出し／書き込み要求信号を生成すると、その増分状態マシンは同時に動作を開始する。増分の増分14は、



現在デコード装置中の命令に示されるメモリ内のアドレスのデコード後、その動作を取り出すためにMCU120またはIOU130のいずれか、Y3でスイッチ125を介してユニット133からそれぞれ取附した番号を生成する。番号が奇り振られれば、取り附された開閉ユニットはアイドル状態に置かれ、要求が応じ消されなければ、閉要ユニットは対応するメモリ上の記憶番組にアクセスし得る。

取組し改善を重ねたものの11月までのビズを決定することによって行われる。そのビズがゼロであれば本来は返金を受ける。そのビズが10以上ある場合は10ダマシに相当するものがある。2次取組によるビズは、1ダマシが0.000、0.000と0.03PF、PFの間にあれば、そのアドレスはメモリにアクセスする。しかしながら、アドレスがPF0.0、PF0.0と0.03PF、PFの間にあれば、10にアクセスする。改善を要請している。他の改善は全て終了されている。本見聞において特に多量のメモリマップが使用できるということが非常に明確である。

再び第2図において、アドレスをデコードすると、そのアドレスが予約領域は230の1ヶ所に割られているかどうかも判別される。予約領域230の1ヶ所に割られているれば、IOU130はMCU120に対する読み出し/書き込みは共に取り消される。

この変換を実行する際には、トレードオフがある。MCU170が低レベルの電力で動作し、書き込みモードになるように要求される時でも、MCU170は低レベルの電力で動作し、読み出し／書き込みモードになる期間電力が増加する。高速度フロー・レートにすることによってシステムの性能を上げれば、ほとんどのアプリケーションでこの取り引きの増加は必要である。

## 53

焼って、早晨明はシステム上のタレント・レートを送  
 りることが出来る仕組みを提案するものである。OPCODE  
 がコードフォーマット、読み出しもしくは書き込みの動作が明  
 らげられることを示しているが、直ちに読み出し/書き  
 込み状態に遷移するようにメモリ並び/IOユニットに命令  
 を送ることによってこのタレント・レートは極めて高く、か  
 してクリティカル・パスから状態遷移時間が取り除かれ

本罪刑について好む実刑例を挙げて具体的に説明  
をしてきたが、本罪刑の目的並びに適用範囲から外れる  
ことなく形状及び様相に於いて様々な変更が可能である  
ことが通常常には理解されることであらう。

側を挙げれば、本発明に基づく技法は半導体素子に設けられたMCUやIOU以外の周辺デバイスを使ってシステムで実行することが可能であり、且つ本発明による技法は多量の資源デバイスを包含するシステムに拡張することができる。

The diagram shows a signal processing system. An input signal  $1/0$  enters a block  $120$  labeled  $\Sigma 220(n \cdot \text{clock})$ . Inside block  $120$ , the signal passes through a delay element  $z^{-1}$  (labeled 127) and then an adder  $+$  (labeled 124). The output of block  $120$  is signal  $142$ . This signal  $142$  is fed into two parallel paths. The first path goes through a block  $131$  labeled  $\Sigma 1/0$  and then an adder  $+$  (labeled 132). The second path goes through a block  $133$  labeled  $1/0$  and then an adder  $+$  (labeled 135). The outputs of these two paths are combined at a summing junction (labeled 136) to produce signal  $140$ . Signal  $140$  is then fed into a block  $150$  labeled  $f(x, y)$ .

一、  
一、

